

Державна служба України з надзвичайних ситуацій
Львівський державний університет безпеки життєдіяльності

Кафедра УПІТТ



Курсова робота

з дисципліни: “Об’єктно-орієнтоване програмування”

на тему:

“Базові принципи об’єктно-орієнтованого програмування”

Індивідуальне завдання: “ОСББ”

Виконав:

кр. гр. КН-21

Анастасія ЧАСТИЛО

Прийняв:

к. т. н.

Олександр ПРИДАТКО

Львів – 2020

Анотація

Курсова робота присвячена розробці програми для спілки ОСББ.

У склад даної роботи входить: програмний код, розробка та опис алгоритму, набір об'єктних моделей опис реалізованих класів, методів, взаємодії класів між собою.

Проводяться усі необхідні роботи програмного забезпечення для створення програми, опис реалізації, кодування, тестування програми.

При написанні курсової роботи використовується об'єктно орієнтована мова програмування Java.

Робота викладена на 33 сторінках, містить 10 рисунків.

ЗМІСТ

1. Анотація	2
2. Зміст	3
3. Вступ	4
4. Розробка та опис алгоритму.....	7
5. Програмна реалізація алгоритму	8
6. Опис реалізованих методів	26
7. Представлення діаграми класів.....	30
8. Висновки	32
9. Список використаної літератури.....	33

ВСТУП

У наш час великого попиту в багатоквартирних будівлях зазнало утворення певних об'єднань для спільного користування співвласниками, утримання та управління, а також для юридичного оформлення їхніх майнових прав на будинок та прибуткову територію. Іншими, більш поширеними, словами таке угруповання можна назвати Об'єднанням співвласників багатоквартирного будинку.

Згідно норм Закону України «Про особливості здійснення права власності у багатоквартирному будинку», ОСББ є однією із форм управління багатоквартирним будинком. Об'єднання створюється на основі права сумісності власності і представляє лише співвласників нерухомості, щоб самостійно вирішувати – хто обслуговуватиме будинок, які будуть внески на його утримання, квартплата, кому здавати в оренду нежитлові приміщення, куди витрачати гроші.

Доцільність створення ОСББ залежить від тих завдань, які мешканці поставляють перед собою, одним з яких може бути контроль за послугами ЖЕКу. ОСББ має право мати розрахункові рахунки, розміщувати депозити в банках, купувати або продавати майно.

ОСББ є одним із ефективних інструментів для залучення співвласниками коштів на проведення енергоефективних заходів та заходів з модернізації будинку. Запровадження таких заходів у багатоквартирних будинках дозволить суттєво зменшити рівень споживання співвласниками енергоресурсів та досягнути економії. На сьогодні відбувається створення ефективної нормативно-правової бази, яка дозволить співвласникам більш ефективно залучати кошти на енергозбереження та модернізацію будинків та здійснювати ефективний облік спожитих ресурсів. Зокрема, вже є прийнятими Закон України «Про Фонд енергоефективності», «Про енергетичну ефективність будівель», «Про комерційний облік теплової енергії та водопостачання», тощо.

Для створення об'єднання скликаються установчі збори. Рішення установчих зборів оформлюється протоколом. В одному багатоквартирному будинку може бути створено тільки одне об'єднання. Власники квартир та нежитлових приміщень у двох і більше багатоквартирних будинках, об'єднаних спільною прибудинковою територією, елементами благоустрою, обладнанням, інженерною інфраструктурою, можуть створити одне об'єднання. Установчі збори об'єднання у новозбудованих багатоквартирних будинках можуть бути проведені після державної реєстрації права власності на більше половини квартир та нежитлових приміщень у такому будинку.

В цій роботі ми більше присвяtimo тему стосовно внесків і платежів співвласників. Стаття 10 Закону України “Про об’єднання співвласників багатоквартирного будинку” передбачає, що до виключної компетенції загальних зборів належить, зокрема, визначення порядку сплати внесків і платежів співвласників. Співвласники зобов’язані сплачувати внески згідно з квитанціями на сплату внесків, сформованими та наданими правлінням Об’єднання.

Отже, для значного полегшення розрахунку податків ОСББ була реалізована дана програма.

За певним прикладом одного із Об’єднань співвласників багатоквартирного будинку, встановленим переліком і розміром внесків співвласників, урахуванням житлової площі, а також інших факторів була реалізована дана програма.

Нашим прикладом є триповерхова будівля, яка складається з дванадцяти квартир площами – 50, 70, 80, 90 м.кв..

Розміри внесків становили:

- внесок на утримання будинку і прибудинкової території – 4,50 грн. на місяць з 1 м. кв. площі приміщення, що належить співвласникові;
- внесок до ремонтного фонду – 0,60 грн. на місяць з 1 м.кв. площі приміщення, що належить співвласникові;
- внесок до резервного фонду – 0,20 грн на місяць з 1 м.кв. площі приміщення, що належить співвласникові.

За допомогою програми ми можемо визначати таку інформацію про будівлю, як: площа кожного поверху, загальна площа будівлі, особисту інформацію про кожного співвласника, кількість жильців в кожній квартирі, на поверсі, а також загальна кількість в будинку, ід кожного співвласника.

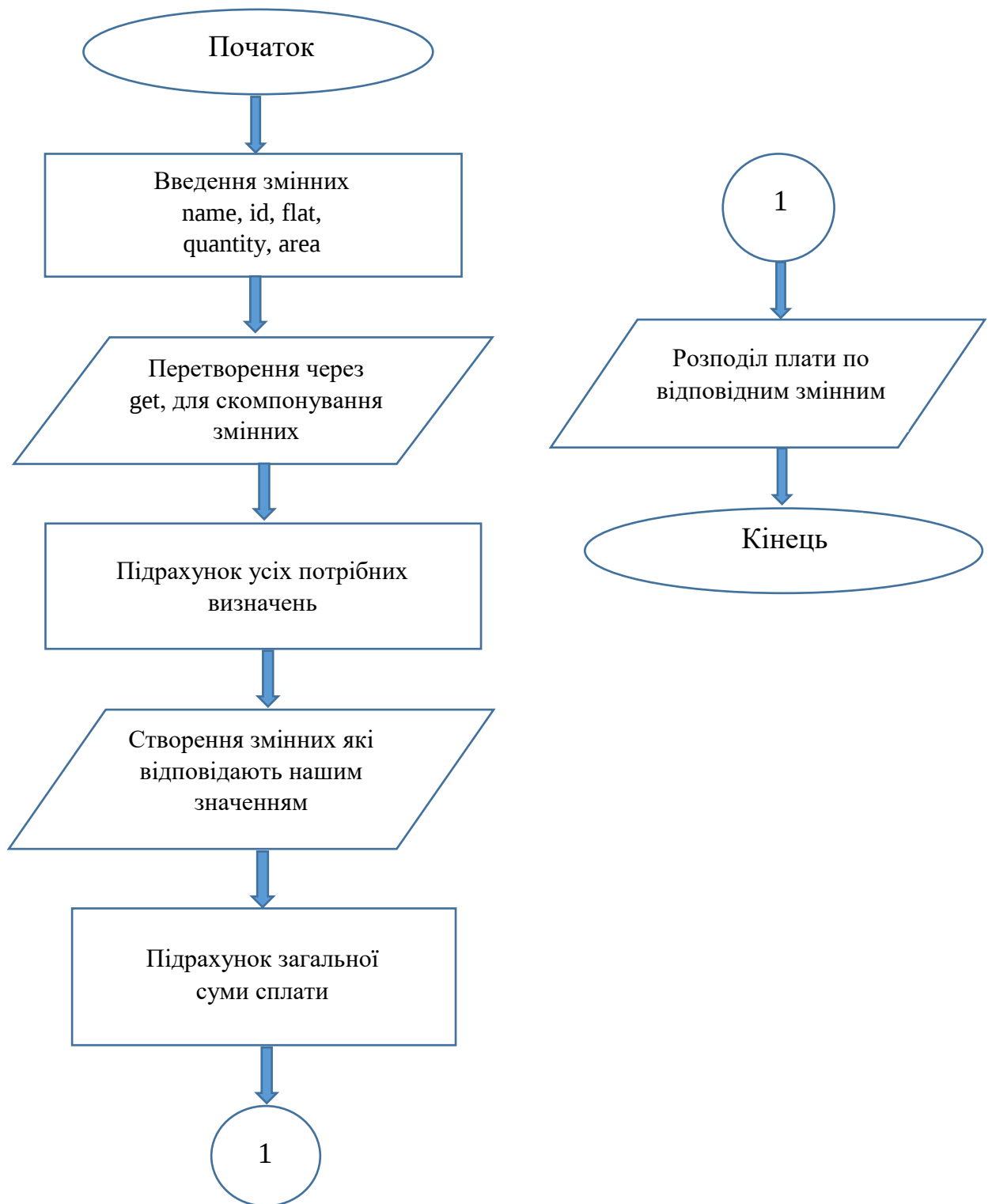
Головною перевагою є те, що програма вираховує місячний внесок для кожного співвласника враховуючи розміри платежів за 1 м.кв., разом із загальною площею квартири. За допомогою спеціальних методів та формул.

За кожним співвласником Об’єднання закріплюється певне ід, яке потім утворює базу усіх жильців по списку. Потім за кожним номером квартири визначається сім’я та кількість проживаючих в ній. Після цього визначається площа кожної квартири, і на сам кінець підраховується сума щомісячного внеску для кожного житлового або нежитлового приміщення.

Ідея написання програми, яка вираховує платежі Об’єднання співвласників багатоквартирного будинку виникла після ознайомлення з певними статтями, посібниками, наказами та законами ОСББ, для полегшення

та оптимізації роботи об'єднання, вирішення проблем з деякими похибками при нарахуванні податку, а також доступності іншим співвласникам.

1. Розробка та опис алгоритму.



2. Програмна реалізація алгоритму.

2.1) Клас NumberApartment – номер квартири з вкладеним класом Main.

```
public class NumberApartment implements Comparable {

    public int family_id;
    public String last_name;

    public NumberApartment (int family_id, String last_name) {
        this.family_id = family_id;
        this.last_name = last_name;
    }

    public int compareTo(Object obj)
    {
        NumberApartment tmp = (NumberApartment)obj;
        if(this.family_id < tmp.family_id)
        {
            return -1;
        }
        else if(this.family_id > tmp.family_id)
        {
            return 1;
        }
        return 0;
    }

    private String flat;

    public NumberApartment (String flat) {
        this.flat = flat;
    }

    public String getFlat() {
        return flat;
    }

    public void setFlat(String flat) {
        this.flat = flat;
    }

    private interface NumberApartmentInfo {
        String getFlat();
    }

    public NumberApartmentInfo getNumberApartmentInfo() {
```

```

        class FamilyNumberApartmentInfo implements
NumberApartmentInfo{
            final String flat = NumberApartment.this.flat;

            public String getFlat() {
                return this.flat;
            }
        }
        return new FamilyNumberApartmentInfo();
    }

    public static void main (String[] args) {
        System.out.println("В будинку №5 такі номери квартир:");

        NumberApartment family1 = new NumberApartment ("11");
        NumberApartment family2 = new NumberApartment ("12");
        NumberApartment family3 = new NumberApartment ("13");
        NumberApartment family4 = new NumberApartment ("14");
        NumberApartment family5 = new NumberApartment ("21");
        NumberApartment family6 = new NumberApartment ("22");
        NumberApartment family7 = new NumberApartment ("23");
        NumberApartment family8 = new NumberApartment ("24");
        NumberApartment family9 = new NumberApartment ("31");
        NumberApartment family10 = new NumberApartment ("32");
        NumberApartment family11 = new NumberApartment ("33");
        NumberApartment family12 = new NumberApartment ("34");

        NumberApartmentInfo number1 =
family1.getNumberApartmentInfo();
        NumberApartmentInfo number2 =
family2.getNumberApartmentInfo();
        NumberApartmentInfo number3 =
family3.getNumberApartmentInfo();
        NumberApartmentInfo number4 =
family4.getNumberApartmentInfo();
        NumberApartmentInfo number5 =
family5.getNumberApartmentInfo();
        NumberApartmentInfo number6 =
family6.getNumberApartmentInfo();
        NumberApartmentInfo number7 =
family7.getNumberApartmentInfo();
        NumberApartmentInfo number8 =
family8.getNumberApartmentInfo();
        NumberApartmentInfo number9 =
family9.getNumberApartmentInfo();
        NumberApartmentInfo number10 =
family10.getNumberApartmentInfo();
        NumberApartmentInfo number11 =
family11.getNumberApartmentInfo();
        NumberApartmentInfo number12 =
family12.getNumberApartmentInfo();
    }
}

```

```

        System.out.println("Квартира №" +number1.getFlat());
        System.out.println("Квартира №" +number2.getFlat());
        System.out.println("Квартира №" +number3.getFlat());
        System.out.println("Квартира №" +number4.getFlat());
        System.out.println("Квартира №" +number5.getFlat());
        System.out.println("Квартира №" +number6.getFlat());
        System.out.println("Квартира №" +number7.getFlat());
        System.out.println("Квартира №" +number8.getFlat());
        System.out.println("Квартира №" +number9.getFlat());
        System.out.println("Квартира №" +number10.getFlat());
        System.out.println("Квартира №" +number11.getFlat());
        System.out.println("Квартира №" +number12.getFlat());
    }
}

```

2.2) Клас Main для класу NumberApartment.

```

import java.util.Arrays;

public class ClassMain {

    public static void main(String[] args) {
        System.out.println("Список сімей будинку №5 за порядковим номером id:");

        NumberApartment[] family = new NumberApartment[12];
        family[0] = new NumberApartment (5856, "Іваненко");
        family[1] = new NumberApartment (1254, "Котова");
        family[2] = new NumberApartment (6354, "Бойко");
        family[3] = new NumberApartment (7586, "Івченко");
        family[4] = new NumberApartment (1125, "Кушнір");
        family[5] = new NumberApartment (4587, "Сторожук");
        family[6] = new NumberApartment (4365, "Богдан");
        family[7] = new NumberApartment (7147, "Хоменко");
        family[8] = new NumberApartment (4869, "Бас");
        family[9] = new NumberApartment (1567, "Павленко");
        family[10] = new NumberApartment (4358, "Бондар");
        family[11] = new NumberApartment (5478, "Іванчук");

        Arrays.sort(family);

        for(int i = 0; i < family.length; i++)
        {
            System.out.println(family[i].family_id + " " +
family[i].last_name);
        }
    }
}

```

2.3) Клас ListFamily – список сімей з вкладеним класом Main.

```
public class ListFamily extends NumberApartment {
    private String name;

    public ListFamily (String flat, String name) {
        super (flat);
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    private interface ListFamilyInfo {
        String getName();
    }

    public ListFamilyInfo getListFamilyInfo() {
        class FamilyListFamilyInfo implements ListFamilyInfo{
            final String name = ListFamily.this.name;

            public String getName() {
                return this.name;
            }
        }
        return new FamilyListFamilyInfo();
    }

    public static void main (String[] args) {
        System.out.println("За загальним списком в будинку №5  
проживають сім'ї:");

        ListFamily family1 = new ListFamily ("11","Іваненко");
        ListFamily family2 = new ListFamily ("12","Котова");
        ListFamily family3 = new ListFamily ("13","Бойко");
        ListFamily family4 = new ListFamily ("14","Івченко");
        ListFamily family5 = new ListFamily ("21","Кушнір");
        ListFamily family6 = new ListFamily ("22","Сторожук");
        ListFamily family7 = new ListFamily ("23","Богдан");
        ListFamily family8 = new ListFamily ("24","Хоменко");
        ListFamily family9 = new ListFamily ("31","Бас");
        ListFamily family10 = new ListFamily ("32","Павленко");
    }
}
```

```

ListFamily family11 = new ListFamily ("33","Бондар");
ListFamily family12 = new ListFamily ("34","Іванчук");

family1.getFlat();
family2.getFlat();
family3.getFlat();
family4.getFlat();
family5.getFlat();
family6.getFlat();
family7.getFlat();
family8.getFlat();
family9.getFlat();
family10.getFlat();
family11.getFlat();
family12.getFlat();

ListFamilyInfo inform1 = family1.getListFamilyInfo();
ListFamilyInfo inform2 = family2.getListFamilyInfo();
ListFamilyInfo inform3 = family3.getListFamilyInfo();
ListFamilyInfo inform4 = family4.getListFamilyInfo();
ListFamilyInfo inform5 = family5.getListFamilyInfo();
ListFamilyInfo inform6 = family6.getListFamilyInfo();
ListFamilyInfo inform7 = family7.getListFamilyInfo();
ListFamilyInfo inform8 = family8.getListFamilyInfo();
ListFamilyInfo inform9 = family9.getListFamilyInfo();
ListFamilyInfo inform10 = family10.getListFamilyInfo();
ListFamilyInfo inform11 = family11.getListFamilyInfo();
ListFamilyInfo inform12 = family12.getListFamilyInfo();

System.out.println("1. В квартирі № "+family1.getFlat()+"
проживає сім'я " +inform1.getName());
System.out.println("2. В квартирі №"+family2.getFlat()+
"проживає сім'я"+ " " +inform2.getName());
System.out.println("3. В квартирі №"+family3.getFlat()+
"проживає сім'я"+ " " +inform3.getName());
System.out.println("4. В квартирі №"+family4.getFlat()+
"проживає сім'я"+ " " +inform4.getName());
System.out.println("5. В квартирі №"+family5.getFlat()+
"проживає сім'я"+ " " +inform5.getName());
System.out.println("6. В квартирі №"+family6.getFlat()+
"проживає сім'я"+ " " +inform6.getName());
System.out.println("7. В квартирі №"+family7.getFlat()+
"проживає сім'я"+ " " +inform7.getName());
System.out.println("8. В квартирі №"+family8.getFlat()+
"проживає сім'я"+ " " +inform8.getName());
System.out.println("9. В квартирі №"+family9.getFlat()+
"проживає сім'я"+ " " +inform9.getName());
System.out.println("10. В квартирі №"+family10.getFlat()+
"проживає сім'я"+ " " +inform10.getName());
System.out.println("11. В квартирі
№"+family11.getFlat()+"проживає сім'я"+ " " +inform11.getName());

```

```

        System.out.println("12. В квартирі №"+family12.getFlat()+"
проживає сім'я"+ " " +inform12.getName());

    }

}

```

2.4 Клас Quantity – кількість з вкладеним класом Main.

```

public class Quantity extends ListFamily {

    private String quantity;

    public Quantity (String flat, String name, String quantity) {
        super(flat, name);
        this.quantity = quantity;
    }

    private interface QuantityInfo {
        String getName();
    }

    public QuantityInfo getQuantityInfo() {
        class FamilyQuantityInfo implements QuantityInfo{
            final String quantity = Quantity.this.quantity;

            public String getName() {
                return this.quantity;
            }
        }
        return new FamilyQuantityInfo();
    }

    public static void main (String[] args) {
        System.out.println("За загальним списком в будинку №5
проживають сім'ї:");

```

```

Quantity family1 = new Quantity ("11","Іваненко", "2");
Quantity family2 = new Quantity ("12","Котова", "1");
Quantity family3 = new Quantity ("13","Бойко", "4");
Quantity family4 = new Quantity ("14","Івченко", "3");
Quantity family5 = new Quantity ("21","Кушнір", "6");
Quantity family6 = new Quantity ("22","Сторожук", "2");
Quantity family7 = new Quantity ("23","Богдан", "4");
Quantity family8 = new Quantity ("24","Хоменко", "1");
Quantity family9 = new Quantity ("31","Бас", "3");
Quantity family10 = new Quantity ("32","Павленко", "5");
Quantity family11 = new Quantity ("33","Бондар", "1");

```

```

Quantity family12 = new Quantity ("34","Іванчук", "2");

family1.getFlat();
family2.getFlat();
family3.getFlat();
family4.getFlat();
family5.getFlat();
family6.getFlat();
family7.getFlat();
family8.getFlat();
family9.getFlat();
family10.getFlat();
family11.getFlat();
family12.getFlat();
family1.getName();
family2.getName();
family3.getName();
family4.getName();
family5.getName();
family6.getName();
family7.getName();
family8.getName();
family9.getName();
family10.getName();
family11.getName();
family12.getName();

QuantityInfo inform1 = family1.getQuantityInfo();
QuantityInfo inform2 = family2.getQuantityInfo();
QuantityInfo inform3 = family3.getQuantityInfo();
QuantityInfo inform4 = family4.getQuantityInfo();
QuantityInfo inform5 = family5.getQuantityInfo();
QuantityInfo inform6 = family6.getQuantityInfo();
QuantityInfo inform7 = family7.getQuantityInfo();
QuantityInfo inform8 = family8.getQuantityInfo();
QuantityInfo inform9 = family9.getQuantityInfo();
QuantityInfo inform10 = family10.getQuantityInfo();
QuantityInfo inform11 = family11.getQuantityInfo();
QuantityInfo inform12 = family12.getQuantityInfo();

System.out.println("1. В квартирі №"+family1.getFlat()+" "+
"проживає сім'я"+ " " +family1.getName()+" " + "з кількістю"+ " "
+family1.quantity+ " людей;");
System.out.println("2. В квартирі №"+family2.getFlat()+" "+
"проживає сім'я"+ " " +family2.getName()+" " + "з кількістю"+ " "
+family2.quantity+ " людей;");
System.out.println("3. В квартирі №"+family3.getFlat()+" "+
"проживає сім'я"+ " " +family3.getName()+" " + "з кількістю"+ " "
+family3.quantity+ " людей;");

```

```

        System.out.println("4. В квартирі №"+family4.getFlat()+" "+
"проживає сім'я"+ " " +family4.getName()+" " + "з кількістю"+ " "
+family4.quantity+ " людей;");
        System.out.println("5. В квартирі №"+family5.getFlat()+" "+
"проживає сім'я"+ " " +family5.getName()+" " + "з кількістю"+ " "
+family5.quantity+ " людей;");
        System.out.println("6. В квартирі №"+family6.getFlat()+" "+
"проживає сім'я"+ " " +family6.getName()+" " + "з кількістю"+ " "
+family6.quantity+ " людей;");
        System.out.println("7. В квартирі №"+family7.getFlat()+" "+
"проживає сім'я"+ " " +family7.getName()+" " + "з кількістю"+ " "
+family7.quantity+ " людей;");
        System.out.println("8. В квартирі №"+family8.getFlat()+" "+
"проживає сім'я"+ " " +family8.getName()+" " + "з кількістю"+ " "
+family8.quantity+ " людей;");
        System.out.println("9. В квартирі №"+family9.getFlat()+" "+
"проживає сім'я"+ " " +family9.getName()+" " + "з кількістю"+ " "
+family9.quantity+ " людей;");
        System.out.println("10. В квартирі №"+family10.getFlat()+"
"+ "проживає сім'я"+ " " +family10.getName()+" " + "з кількістю"+ " "
+family10.quantity+ " людей;");
        System.out.println("11. В квартирі №"+family11.getFlat()+"
"+ "проживає сім'я"+ " " +family11.getName()+" " + "з кількістю"+ " "
+family11.quantity+ " людей;");
        System.out.println("12. В квартирі №"+family12.getFlat()+"
"+ "проживає сім'я"+ " " +inform12.getName()+" " + "з кількістю"+ " "
+family12.quantity+ " людей.");

```

```

    }

```

```

}

```

2.5) Клас – Suma - сума з вкладеним класом Main.

```

public class Suma{
    int fam1;
    int fam2;
    int fam3;
    int fam4;
    int fam5;
    int fam6;
    int fam7;
    int fam8;
    int fam9;
    int fam10;
    int fam11;
    int fam12;

```

```

    public Suma(int fam1, int fam2, int fam3, int fam4, int fam5, int
fam6,
                int fam7, int fam8, int fam9, int fam10, int fam11,
int fam12) {
    this.fam1 = fam1;
    this.fam2 = fam2;
    this.fam3 = fam3;
    this.fam4 = fam4;
    this.fam5 = fam5;
    this.fam6 = fam6;
    this.fam7 = fam7;
    this.fam8 = fam8;
    this.fam9 = fam9;
    this.fam10 = fam10;
    this.fam11 = fam11;
    this.fam12 = fam12;
}

    public Suma(Object object, Object object2, Object object3) {
        // TODO Auto-generated constructor stub
    }

    void Flat1 () {
        int sum1;
        sum1 = fam1 + fam2 + fam3+ fam4;
        System.out.println("Загальна кількість проживаючих на 1
поверхі :" +sum1);
    }

    void Flat2 () {
        int sum2;
        sum2 = fam5 + fam6 + fam7+ fam8;
        System.out.println("Загальна кількість проживаючих на 2
поверхі :" +sum2);
    }

    void Flat3 () {
        int sum3;
        sum3 = fam9 + fam10 + fam11+ fam12;
        System.out.println("Загальна кількість проживаючих на 3
поверхі :" +sum3);
    }

    void AllSum() {
        int AllSum = fam1 + fam2 + fam3+ fam4 + fam5 + fam6+ fam7 +
fam8 + fam9+ fam10 + fam11 + fam12;
        System.out.println("Загальна кількість проживаючих у будинку
: "+AllSum);
    }

    public static void main (String[] args) {

```

```

        System.out.println("Кількість мешканців по поверхах:");
        Suma fam = new Suma (null, null, null);
        fam.fam1= 2;
        fam.fam2= 1;
        fam.fam3= 4;
        fam.fam4= 3;
        fam.fam5= 6;
        fam.fam6= 2;
        fam.fam7= 4;
        fam.fam8= 1;
        fam.fam9= 3;
        fam.fam10= 5;
        fam.fam11= 1;
        fam.fam12= 2;

        fam.Flat1();
        fam.Flat2();
        fam.Flat3();
        fam.AllSum();
    }
}

```

2.6) Клас NumberApartment – квартири з вкладеним класом Main.

```

public class HouseArea extends Quantity{

    public HouseArea(String flat, String name, String quantity) {
        super(flat, name, quantity);
        // TODO Auto-generated constructor stub
    }

    int area1 = 50;
    int area2 = 70;
    int area3 = 80;
    int area4 = 90;
    private int house1;
    private int house2;
    private int house3;
    private int house4;
    private int house5;
    private int house6;
    private int house7;
    private int house8;
    private int house9;
    private int house10;
    private int house11;
    private int house12;
    private int stairs = 7;

    public int getHouse1() {
        return house1;
    }
}

```

```

}

public void setHouse1(int house1) {
    this.house1 = house1;
}

public int getHouse2() {
    return house2;
}

public void setHouse2(int house2) {
    this.house2 = house2;
}

public int getHouse3() {
    return house3;
}

public void setHouse3(int house3) {
    this.house3 = house3;
}

public int getHouse4() {
    return house4;
}

public void setHouse4(int house4) {
    this.house4 = house4;
}

public int getHouse5() {
    return house5;
}

public void setHouse5(int house5) {
    this.house5 = house5;
}

public int getHouse6() {
    return house6;
}

public void setHouse6(int house6) {
    this.house6 = house6;
}

public int getHouse7() {
    return house7;
}

public void setHouse7(int house7) {

```

```
        this.house7 = house7;
    }

    public int getHouse8() {
        return house8;
    }

    public void setHouse8(int house8) {
        this.house8 = house8;
    }

    public int getHouse9() {
        return house9;
    }

    public void setHouse9(int house9) {
        this.house9 = house9;
    }

    public int getHouse10() {
        return house10;
    }

    public void setHouse10(int house10) {
        this.house10 = house10;
    }

    public int getHouse11() {
        return house11;
    }

    public void setHouse11(int house11) {
        this.house11 = house11;
    }

    public int getHouse12() {
        return house12;
    }

    public void setHouse12(int house12) {
        this.house12 = house12;
    }

    public int getStairs() {
        return stairs;
    }

    public void setStairs(int stairs) {
        this.stairs = stairs;
    }
}
```

```

    void Area () {
        house1 = house5 = house9 = area1;
        house2 = house6 = house10 = area2;
        house3 = house7 = house11 = area3;
        house4 = house8 = house12 = area4;
    }

    double SumArea1Flat(int house1, int house2, int house3, int
house4, int stairs) {
        double SumArea1Flat = house1+house2+house3+house4+stairs;
        return SumArea1Flat;
    }

    double HouseArea (int house1, int house2, int house3, int house4,
int stairs) {
        double HouseArea = (house1+house2+house3+house4+stairs)*4;
        return HouseArea;
    }

    public static void main (String[] args) {
        System.out.println("Квадратура:");
        Quantity family1 = new Quantity ("11", "Іваненко", "2");
        Quantity family2 = new Quantity ("12", "Котова", "1");
        Quantity family3 = new Quantity ("13", "Бойко", "4");
        Quantity family4 = new Quantity ("14", "Івченко", "3");
        Quantity family5 = new Quantity ("21", "Кушнір", "6");
        Quantity family6 = new Quantity ("22", "Сторожук", "2");
        Quantity family7 = new Quantity ("23", "Богдан", "4");
        Quantity family8 = new Quantity ("24", "Хоменко", "1");
        Quantity family9 = new Quantity ("31", "Бас", "3");
        Quantity family10 = new Quantity ("32", "Павленко", "5");
        Quantity family11 = new Quantity ("33", "Бондар", "1");
        Quantity family12 = new Quantity ("34", "Іванчук", "2");

        family1.getFlat();
        family2.getFlat();
        family3.getFlat();
        family4.getFlat();
        family5.getFlat();
        family6.getFlat();
        family7.getFlat();
        family8.getFlat();
        family9.getFlat();
        family10.getFlat();
        family11.getFlat();
        family12.getFlat();
        HouseArea ha = new HouseArea (null, null, null);
        ha.area1 = 50;
        ha.area2 = 70;
        ha.area3 = 80;
        ha.area4 = 90;
        ha.stairs = 7;
    }
}

```

```

        ha.house1 = ha.house5 = ha.house9 = ha.area1;
        ha.house2 = ha.house6 = ha.house10 = ha.area2;
        ha.house3 = ha.house7 = ha.house11 = ha.area3;
        ha.house4 = ha.house8 = ha.house12 = ha.area4;
        ha.SumArea1Flat(ha.area1, ha.area2, ha.area3,
ha.area4, ha.stairs);
        ha.HouseArea(ha.area1, ha.area2, ha.area3, ha.area4,
ha.stairs);

        System.out.println("Квартира № "+family1.getFlat()+" має
площу " + ha.house1+ " кв.м.");
        System.out.println("Квартира № "+family2.getFlat()+" має
площу " + ha.house2+ " кв.м.");
        System.out.println("Квартира № "+family3.getFlat()+" має
площу " + ha.house3+ " кв.м.");
        System.out.println("Квартира № "+family4.getFlat()+" має
площу " + ha.house4+ " кв.м.");
        System.out.println("Квартира № "+family5.getFlat()+" має
площу " + ha.house5+ " кв.м.");
        System.out.println("Квартира № "+family6.getFlat()+" має
площу " + ha.house6+ " кв.м.");
        System.out.println("Квартира № "+family7.getFlat()+" має
площу " + ha.house7+ " кв.м.");
        System.out.println("Квартира № "+family8.getFlat()+" має
площу " + ha.house8+ " кв.м.");
        System.out.println("Квартира № "+family9.getFlat()+" має
площу " + ha.house9+ " кв.м.");
        System.out.println("Квартира № "+family10.getFlat()+" має
площу " + ha.house10+ " кв.м.");
        System.out.println("Квартира № "+family11.getFlat()+" має
площу " + ha.house11+ " кв.м.");
        System.out.println("Квартира № "+family12.getFlat()+" має
площу " + ha.house12+ " кв.м.");
        System.out.println("Загальна площа одного поверху = "
+ha.SumArea1Flat(ha.area1, ha.area2, ha.area3, ha.area4, ha.stairs)+ "
м.кв.");
        System.out.println("Площа будинку = "
+ha.HouseArea(ha.area1, ha.area2, ha.area3, ha.area4, ha.stairs)+ "
м.кв.");

    }
}

```

2.7) Клас Payment – оплата з вкладеним класом Main.

```

public class Payment extends HouseArea {

    private double territory = 4.50;
    private double repairs = 0.60;
    private double reserve = 0.20;

```

```

    Payment(String flat, String name, String quantity, double
territory, double repairs, double reserve) {
    super(flat, name, quantity);
    this.territory = territory;
    this.repairs = repairs;
    this.reserve = reserve;
}

    double payment1 () {
        double pay1, pay2, pay3, pay4 ;
        pay1 = (territory+repairs+reserve)*area1;
        return pay1;
    }
    double payment2 () {
        double pay2;
        pay2 = (territory+repairs+reserve)*area2;
        return pay2;
    }

    double payment3 () {
        double pay3;
        pay3 = (territory+repairs+reserve)*area3;
        return pay3;
    }

    double payment4 () {
        double pay4;
        pay4 = (territory+repairs+reserve)*area4;
        return pay4;
    }

    public double getTerritory() {
        return territory;
    }
    public void setTerritory(double territory) {
        this.territory = territory;
    }
    public double getRepairs() {
        return repairs;
    }
    public void setRepairs(double repairs) {
        this.repairs = repairs;
    }
    public double getReserve() {
        return reserve;
    }
    public void setReserve(double reserve) {
        this.reserve = reserve;
    }

    public static void main(String[] args) {
        Payment pay = new Payment(null, null, null, 0, 0, 0);

```

```

        pay.setTerritory(4.50);
        pay.setRepairs(0.60);
        pay.setReserve(0.20);
        pay.payment1();
        pay.payment2();
        pay.payment3();
        pay.payment4();
        pay.area1= 50;
        pay.area2= 70;
        pay.area3= 80;
        pay.area4= 90;

        System.out.println("Виплати за місяць по квадратурі:");
        System.out.println("Якщо квартира = " +pay.area1+ " кв.м,
тоді внесок за місяць = " + pay.payment1()+ " грн.");
        System.out.println("Якщо квартира = " +pay.area2+ " кв.м,
тоді внесок за місяць = " + pay.payment2()+ " грн.");
        System.out.println("Якщо квартира = " +pay.area3+ " кв.м,
тоді внесок за місяць = " + pay.payment3()+ " грн.");
        System.out.println("Якщо квартира = " +pay.area4+ " кв.м,
тоді внесок за місяць = " + pay.payment4()+ " грн.");

    }
}

```

2.8) Клас Report – звіт з вкладеним класом Main.

```

public class Report extends Payment{

    private double pay1 = 265.0;
    private double pay2 = 371.0;
    private double pay3 = 424.0;
    private double pay4 = 477.0;

    Report(String flat, String name, String quantity, double territory,
double repairs, double reserve) {
        super(flat, name, quantity, territory, repairs, reserve);
        // TODO Auto-generated constructor stub
    }

    public double getPay1() {
        return pay1;
    }

    public void setPay1(double pay1) {
        this.pay1 = pay1;
    }

    public double getPay2() {
        return pay2;
    }

    public void setPay2(double pay2) {
        this.pay2 = pay2;
    }
}

```

```

}
public double getPay3() {
    return pay3;
}
public void setPay3(double pay3) {
    this.pay3 = pay3;
}
public double getPay4() {
    return pay4;
}
public void setPay4(double pay4) {
    this.pay4 = pay4;
}

public static void main (String[] args) {
    Report rpt = new Report (null, null, null, 0, 0, 0);
    rpt.pay1= 265.0;
    rpt.pay2= 371.0;
    rpt.pay3= 424.0;
    rpt.pay4= 477.0;

    Quantity family1 = new Quantity ("11","Іваненко", "2");
    Quantity family2 = new Quantity ("12","Котова", "1");
    Quantity family3 = new Quantity ("13","Бойко", "4");
    Quantity family4 = new Quantity ("14","Івченко",
"3");

    Quantity family5 = new Quantity ("21","Кушнір", "6");
    Quantity family6 = new Quantity ("22","Сторожук",
"2");

    Quantity family7 = new Quantity ("23","Богдан", "4");
    Quantity family8 = new Quantity ("24","Хоменко",
"1");

    Quantity family9 = new Quantity ("31","Бас", "3");
    Quantity family10 = new Quantity ("32","Павленко",
"5");

    Quantity family11 = new Quantity ("33","Бондар", "1");
    Quantity family12 = new Quantity ("34","Іванчук",
"2");

    System.out.println("Місячний внесок:");
    System.out.println("Мешканці квартири № "
+family1.getFlat()+ " мають виплатити "+rpt.pay1+ " грн.");
    System.out.println("Мешканці квартири № "
+family2.getFlat()+ " мають виплатити "+rpt.pay2+ " грн.");
    System.out.println("Мешканці квартири № "
+family3.getFlat()+ " мають виплатити "+rpt.pay3+ " грн.");
    System.out.println("Мешканці квартири № "
+family4.getFlat()+ " мають виплатити "+rpt.pay4+ " грн.");
    System.out.println("Мешканці квартири № "
+family5.getFlat()+ " мають виплатити "+rpt.pay1+ " грн.");
    System.out.println("Мешканці квартири № "
+family6.getFlat()+ " мають виплатити "+rpt.pay2+ " грн.");

```

```

        System.out.println("Мешканці квартири № "
+family7.getFlat()+ " мають виплатити "+rpt.pay3+ " грн.");
        System.out.println("Мешканці квартири № "
+family8.getFlat()+ " мають виплатити "+rpt.pay4+ " грн.");
        System.out.println("Мешканці квартири № "
+family9.getFlat()+ " мають виплатити "+rpt.pay1+ " грн.");
        System.out.println("Мешканці квартири № "
+family10.getFlat()+ " мають виплатити "+rpt.pay2+ " грн.");
        System.out.println("Мешканці квартири № "
+family11.getFlat()+ " мають виплатити "+rpt.pay3+ " грн.");
        System.out.println("Мешканці квартири № "
+family12.getFlat()+ " мають виплатити "+rpt.pay4+ " грн.");
    }
}

```

3. Опис реалізованих методів.

Клас NumberApartment.

З самого початку клас NumberApartment імплементує клас Comparable, для того, щоб в класі Main спрацював згенерований метод Sort. Створивши змінні id (family_id) та прізвища (last_name), генеруємо конструктор, а також метод за допомогою if та return, який через клас Main відсортує усі id у порядку зростання.

Вводимо наступну зміну прайвіт квартира (flat), створюємо до неї get/set, імплементуємо та у вкладеному Main класі створюємо список, який буде виводити усі існуючі номери квартир у будинку.

```
<terminated> NumberApartment [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (15 мая 2020 г., 00:00:15 – 00:00:17)
В будинку №5 такі номери квартир:
Квартира №11
Квартира №12
Квартира №13
Квартира №14
Квартира №21
Квартира №22
Квартира №23
Квартира №24
Квартира №31
Квартира №32
Квартира №33
Квартира №34
```

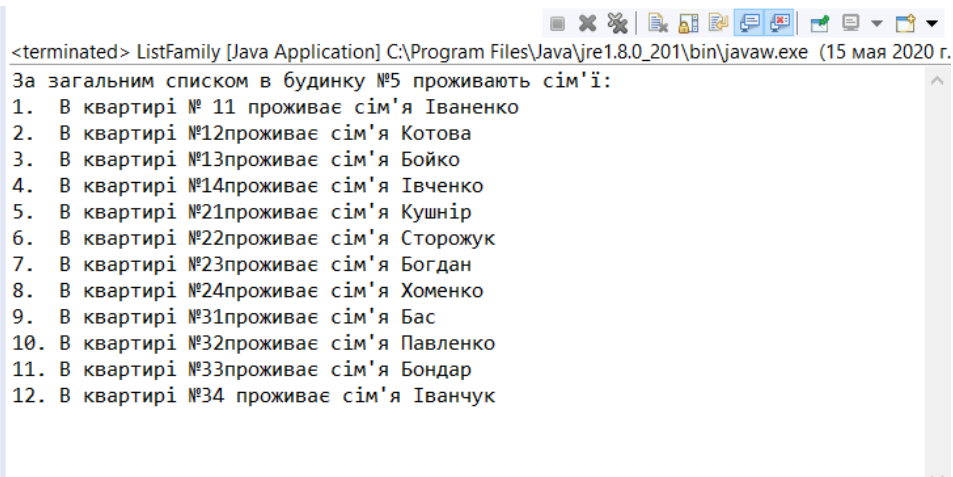
Клас ClassMain.

Даний клас імпортується з java.util.Arrays, для спрацювання метода сортування id. Створивши список усіх сімей, які проживають в будинку, та надавши їм id номер, через метод Sort виводимо у консоль відсортований за зростанням список проживаючих.

```
<terminated> ClassMain [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (15 мая 2020 г.)
Список сімей будинку №5 за порядковим номером id:
1125 Кушнір
1254 Котова
1567 Павленко
4358 Бондар
4365 Богдан
4587 Сторожук
4869 Бас
5478 Іванчук
5856 Іваненко
6354 Бойко
7147 Хоменко
7586 Івченко
```

Клас ListFamily.

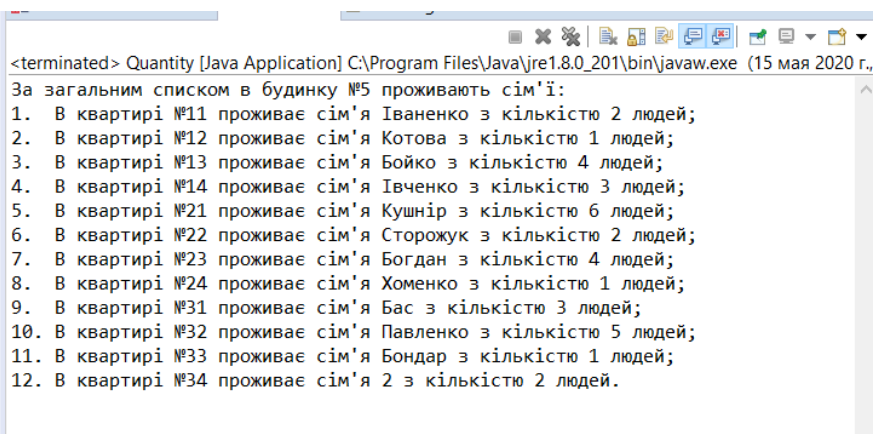
Клас ListFamily наслідує попередній клас NumberApartment. Створюємо прайвіт змінну ім'я name, та через метод за допомогою гетерів, у вкладеному класі Main створюємо список якому підпорядковується прізвища жильців та номер квартири.



```
<terminated> ListFamily [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (15 мая 2020 г.)
За загальним списком в будинку №5 проживають сім'ї:
1. В квартирі № 11 проживає сім'я Іваненко
2. В квартирі №12 проживає сім'я Котова
3. В квартирі №13 проживає сім'я Бойко
4. В квартирі №14 проживає сім'я Івченко
5. В квартирі №21 проживає сім'я Кушнір
6. В квартирі №22 проживає сім'я Сторожук
7. В квартирі №23 проживає сім'я Богдан
8. В квартирі №24 проживає сім'я Хоменко
9. В квартирі №31 проживає сім'я Бас
10. В квартирі №32 проживає сім'я Павленко
11. В квартирі №33 проживає сім'я Бондар
12. В квартирі №34 проживає сім'я Іванчук
```

Клас Quantity.

Наслідує клас ListFamily. Створивши стрічку кількість (quantity), генеруємо конструктор для нової змінної, та який підтягує попередні. Через метод за допомогою гетерів, у вкладеному класі Main створюємо список якому підпорядковується прізвища жильців, номер квартири та кількість в ній проживаючих.



```
<terminated> Quantity [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (15 мая 2020 г.)
За загальним списком в будинку №5 проживають сім'ї:
1. В квартирі №11 проживає сім'я Іваненко з кількістю 2 людей;
2. В квартирі №12 проживає сім'я Котова з кількістю 1 людей;
3. В квартирі №13 проживає сім'я Бойко з кількістю 4 людей;
4. В квартирі №14 проживає сім'я Івченко з кількістю 3 людей;
5. В квартирі №21 проживає сім'я Кушнір з кількістю 6 людей;
6. В квартирі №22 проживає сім'я Сторожук з кількістю 2 людей;
7. В квартирі №23 проживає сім'я Богдан з кількістю 4 людей;
8. В квартирі №24 проживає сім'я Хоменко з кількістю 1 людей;
9. В квартирі №31 проживає сім'я Бас з кількістю 3 людей;
10. В квартирі №32 проживає сім'я Павленко з кількістю 5 людей;
11. В квартирі №33 проживає сім'я Бондар з кількістю 1 людей;
12. В квартирі №34 проживає сім'я 2 з кількістю 2 людей.
```

Клас Suma.

У класі Suma ми створюємо змінні для кожної сім'ї, які проживають у будинку. Створюємо методи, які будуть підраховувати загальну кількість людей на кожному поверсі, та людей які в загальному проживають в будинку. У вкладеному класі надаємо кожній змінній значення, яке відповідає кількості людей в кожній квартирі.

```
<terminated> Suma [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (15 мая 2020 г., 00
Кількість мешканців по поверхах:
Загальна кількість проживаючих на 1 поверсі :10
Загальна кількість проживаючих на 2 поверсі :13
Загальна кількість проживаючих на 3 поверсі :11
Загальна кількість проживаючих у будинку : 34
```

Клас HouseArea.

Клас HouseArea наслідує попередній клас який визначав кількість жильців. В цьому класі ми створюємо змінні усіх площ квартир, та надаємо їм певного значення. Створюємо методи, які підраховують площу на одному поверсі, та повністю площі будинку. Через вкладений клас визначаємо, яку площу мають усі квартири у будинку.

```
<terminated> HouseArea [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (15 мая 2020
Квадратура:
Квартира № 11 має площу 50 кв.м.
Квартира № 12 має площу 70 кв.м.
Квартира № 13 має площу 80 кв.м.
Квартира № 14 має площу 90 кв.м.
Квартира № 21 має площу 50 кв.м.
Квартира № 22 має площу 70 кв.м.
Квартира № 23 має площу 80 кв.м.
Квартира № 24 має площу 90 кв.м.
Квартира № 31 має площу 50 кв.м.
Квартира № 32 має площу 70 кв.м.
Квартира № 33 має площу 80 кв.м.
Квартира № 34 має площу 90 кв.м.
Загальна площа одного поверху = 297.0 м.кв.
Площа будинку = 1188.0 м.кв.
```

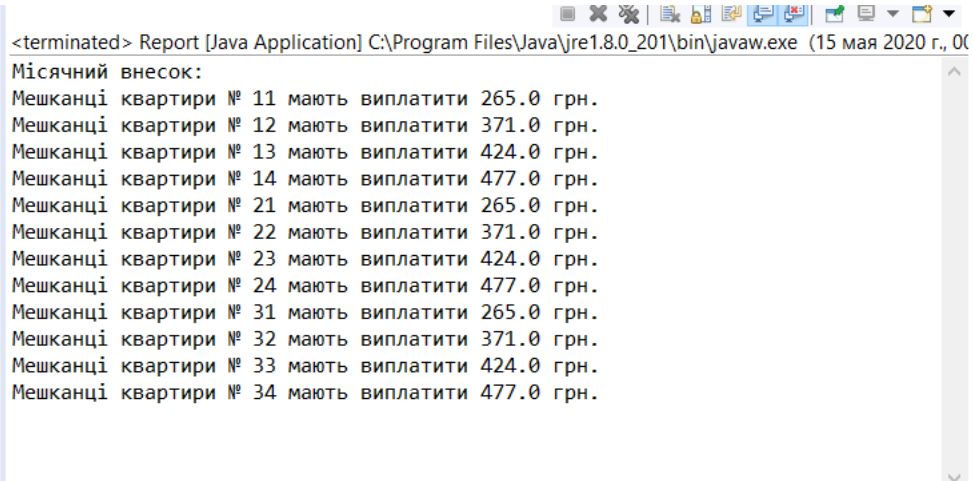
Клас Payment.

У даному класі створенні змінні, які відповідають певним внескам на 1 м.кв.. За допомогою цих змінних та унаслідуваних створюємо методи, які будуть підраховувати щомісячну плату для квартир з певною площею.

```
Problems JavaDoc Console Coverage
<terminated> Payment [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (15 мая 2020 г.,
Виплати за місяць по квадратурі:
Якщо квартира = 50 кв.м, тоді внесок за місяць = 265.0 грн.
Якщо квартира = 70 кв.м, тоді внесок за місяць = 371.0 грн.
Якщо квартира = 80 кв.м, тоді внесок за місяць = 424.0 грн.
Якщо квартира = 90 кв.м, тоді внесок за місяць = 477.0 грн.
```

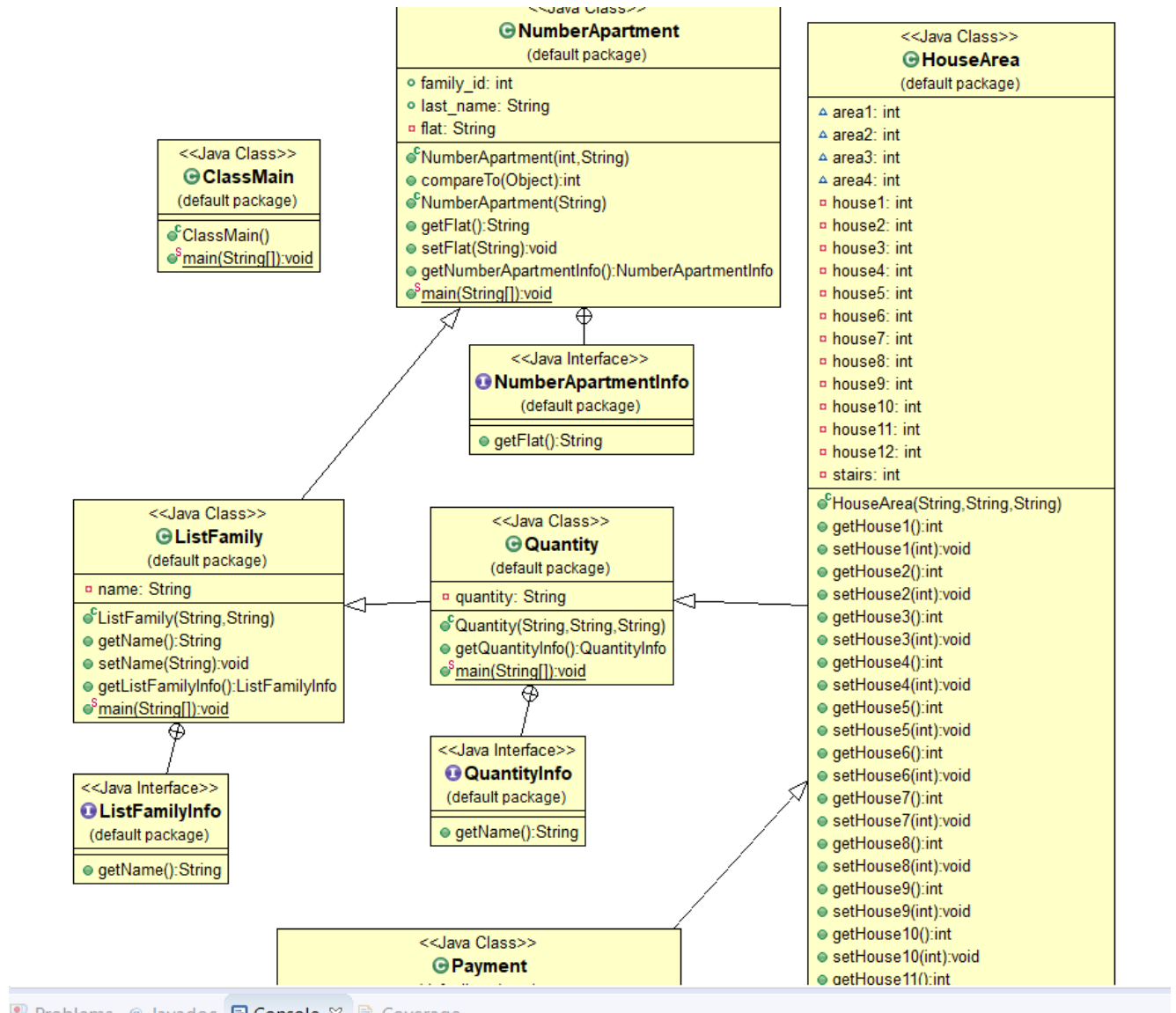
Клас Report.

Даний клас створює звіт, який розділяє відповідну плату за місяць по кожному номеру квартири.



```
<terminated> Report [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (15 мая 2020 г., 00:00)
Місячний внесок:
Мешканці квартири № 11 мають виплатити 265.0 грн.
Мешканці квартири № 12 мають виплатити 371.0 грн.
Мешканці квартири № 13 мають виплатити 424.0 грн.
Мешканці квартири № 14 мають виплатити 477.0 грн.
Мешканці квартири № 21 мають виплатити 265.0 грн.
Мешканці квартири № 22 мають виплатити 371.0 грн.
Мешканці квартири № 23 мають виплатити 424.0 грн.
Мешканці квартири № 24 мають виплатити 477.0 грн.
Мешканці квартири № 31 мають виплатити 265.0 грн.
Мешканці квартири № 32 мають виплатити 371.0 грн.
Мешканці квартири № 33 мають виплатити 424.0 грн.
Мешканці квартири № 34 мають виплатити 477.0 грн.
```

4. Представлення діаграми класів.





ВИСНОВКИ

За допомогою реалізації даної програми маємо можливість дізнатись багато чого про спілку ОСББ, до прикладу:

- визначати рівень квартирної плати;
- визначати розмір внесків на утримання будинку;
- визначати черговість проведення поточного ремонту і контролювати його якість;
- визначати структуру управління будинком;
- контролювати безпеку проживання у будинку та використання прибудинкової території та ін.

На базі цієї інформації і виникла ідея реалізації програми, яка буде визначати вартість певних внесків та контролювати дії співвласників.

Перед написанням програмного коду був створений певний алгоритм, який би надалі відповідав схемі самої програми.

За допомогою об'єктно-орієнтованої мови програмування Java та навичок програмування вдалось відтворити тему індивідуального завдання використовуючи, вивчені раніше, способи створення класів, різних методів, таких як: поліморфізм, унаслідування, інкапсуляція та ін.

Реалізація даної програми може стати у нагоді головним керівникам будь-якого ОСББ, бо виконує такі функції як:

- визначення співвласника будинку по id;
- відтворення площі будинку у загальному, та по поверху;
- підрахунок кількості жильців у будинку, та у кожній квартирі;
- визначення в якій квартирі проживає яка сім'я;
- підрахунок сплати внесків за місяць, та розподіл кожної суми по потрібним квартирам.

Список використаної літератури (джерел)

1. <https://proosbb.info/node/70>
2. https://docs.google.com/document/d/1S89ZAy48huJtXKeR_aGIYeNNja3CrzPITlk1bAyjwGQ/edit#
3. <https://docs.google.com/document/d/18kCRrOswNEyuVPZJmW54278c6WMEoWFRmKWklprSuMQ/edit>
4. https://lb.ua/blog/mariya_osipchuk/360760_shcho_take_osbb_i_chogo_varto_boyatisya.html
5. <https://drive.google.com/file/d/1rCGMyk8NtOymulrEJPVGrX6yIAXFgCh/view>
6. <https://drive.google.com/file/d/1So5bZ2DKnVLLXAPgxbNLq94w7BAcDUqM/view>